Semi-Supervised Learning with Auxiliary Evaluation Component for Large Scale e-Commerce Text Classification

Mingkuan Liu, Musen Wen, Selcuk Kopru, Xianjing Liu, Alan Lu
eBay Inc.,
2145 Hamilton Avenue, San Jose, CA, 95032, USA
{mingkliu,mwen,skopru,xianjliu,alalu}@ebay.com

Abstract

The lack of high-quality labeled training data has been one of the critical challenges facing many industrial machine learning tasks. To tackle this challenge, in this paper, we propose a semi-supervised learning method to utilize unlabeled data and user feedback signals to improve the performance of ML models. The method employs a primary model $Main$ and an auxiliary evaluation model $Eval$, where $Main$ and $Eval$ models are trained iteratively by automatically generating labeled data from unlabeled data and/or users feedback signals. The proposed approach is applied to different text classification tasks. We report results on both the publicly available Yahoo! Answers dataset and our e-commerce product classification dataset. The experimental results show that the proposed method reduces the classification error rate by 4% and up to 15% across various experimental setups and datasets. A detailed comparison with other semi-supervised learning approaches is also presented later in the paper. The results from various text classification tasks demonstrate that our method outperforms those developed in previous related studies.

1 Introduction

There are many ways to improve the performance of a machine learning model. Improving the training data is one such method. Obtaining high-quality training data, such as human labeled data, is usually expensive and time-consuming. Many machine learning systems use unlabeled data or a mixture of labeled and unlabeled data for training because it is cheaper and easier to collect enormous amounts of unlabeled data. Industry-deployed machine learning systems that serve millions of users generate vast amounts of unlabeled data and noisy user feedback signals every day. Those data and signals are very important and can be utilized in the training of real-world machine learning systems.

In this paper, we propose a new semi-supervised learning method with a feedback loop to leverage vast amounts of unlabeled data and feedback signals. In particular, we train two machine learning models iteratively. The main model, which is represented as $Main$, performs the main task at runtime. The auxiliary model, which is represented as $Eval$, works offline and it estimates the correctness of the $Main$ models output. The information available to the auxiliary model $Eval$ is much richer than the run-time model $Main$. Extra data, such as user feed-back data and session context information, can be used when training the auxiliary model. The idea is to control the false positive rate of $Eval$ to produce high-quality, automatically labeled data from unlabeled data. The entire process runs iteratively and the performance of both models is improved in an iterative manner. The assumption of run-time $Main$ model has much fewer available information is due to the business logic flow and/or UX design constraints which limit the run-time $Main$ model to collect richer features in some industry setups.

In this paper, we use text classification experiments to illustrate the proposed approach. However, this semi-supervised learning approach can also be applied to other machine learning tasks, such as machine translation and search relevance.

Different semi-supervised learning approaches have been previously proposed to leverage unlabeled data, including Blum and Mitchell (1998), Weiss et al. (2016), Goodfellow et al. (2014) and

Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pages 68–76
Melbourne, Australia July 19, 2018.

Cohn et al. (1996). Experimental results on the public Yahoo! Answers dataset and on a new public e-commerce dataset for product classification demonstrate the advantages and potential of the proposed framework compared with previous work.

The contributions of this work are as follows:

- A new semi-supervised learning method with the introducing of an auxiliary evaluation component, and
- A scalable, cost-effective and efficient way to convert vast amounts of unlabeled data into high quality labeled data for supervised training purposes.

In section 2, we present the details of the proposed semi-supervised learning approach in the context of text classification tasks. In section 3, the theoretical analysis of the proposed method is provided. Next, we give an overview of related works and highlight their differences compared to our approach. Section 5 defines various experimental setups and presents the results for two different datasets. Finally, we present the papers conclusions in section 6.

2 Proposed Method

2.1 Modeling Approach

The suggested method is based on a few observations from real-world machine learning systems.

- The main model $Main$ that serves online users may have limited information available at prediction time due to the business logic flow or UX design constraints in industry setups. Therefore, the predictions from the $Main$ component may contain errors and cannot be used directly as labeled data for model training purposes.
- The evaluation model $Eval$ runs offline. Hence, it can access much richer information than the main task component without those constraints. User feedback data, user behavioral data, prior and post task session history data, and the knowledge base about the users main task constitute richer offline information. All those data helps the $Eval$ model to reliably estimate whether Mains output is acceptable or not.
- Large-scale supervised machine learning methods typically need a larger amount of labeled training data for a good performance. However, in reality, it is very expensive to manually label millions of training data. On the other hand, large scale machine learning systems serving millions of users are generating millions of unlabeled data and user feedback signals every day that is not effectively utilized.

The main idea of the proposed approach is to train and deploy two parallel machine learning models. The first model $Main$ is used to serve live user requests for the main task. The second machine learning model $Eval$ is used as an offline model to estimate the accuracy of $Main$s prediction. The $Eval$ model utilizes additional signals, such as user feedback signals, system logs that are related to user sessions, the output confidence scores from Main, etc. An EM-style iterative process is applied to train $Main$ and $Eval$ in a repeated manner. High-quality, automatically labeled data are extracted from unlabeled data by controlling the false positive rate and the false negative rate in $Eval$. Mathematical analysis (section 3) and experiments (section 5) on different datasets show that after multiple iterations, the accuracy of Main can be improved substantially.

Algorithm 1 shows the details of the proposed semi-supervised learning algorithm. The input to the algorithm is an initial small set of labeled data L, a large set of unlabeled data U and an optional set of user feedback data F. The algorithm leverages the auxiliary evaluation model $Eval$ and the optional user feedback data F to produce high-quality, automatically labeled data AL from a large amount of unlabeled data U. Once the labeled data AL is extracted and added to the training corpus C, we use a shallow neural network[1] to train model $Main$ for the text classification tasks.

2.2 Auxiliary Evaluation Model

The auxiliary evaluation model $Eval$ is a binary classifier that predicts whether the automatic label is correct or not. It is trained using gradient boosting[2]. If the false positive rate of the eval-

[1] We use the FastText library Joulin et al. (2016) for the shallow neural network implementation.

[2] We use the XGboost library Chen and Guestrin (2016) for gradient boosting.

Algorithm 1 Proposed Algorithm

Given:
Labeled dataset $L : \{l_i\}$
Unlabeled dataset $U : \{u_i\}$
Optional feedback data $F : \{f_i\}$
$Main$ = trainMainModel(L)
$fpRate$ = false positive rate threshold
for $k = 1$ **to** $MaxIter$ **do**
  $Eval$ = trainEvalModel($Main$, L, F)
  AL = emptyList()
  for u_i **in** U **do**
    PC_i = predClass(M, u_i)
    $Score(PC_i)$ = evalScore(E, u_i, f_i, PC_i)
    if $Score(PC_i) > (1 - fpRate)$ **then**
      AL.append(u_i, PC_i)
    end if
  end for
  $C = L + AL$
  M = trainMainModel(C)
end for

uation model $Eval$ is controlled at a low threshold, vast amounts of high-quality, automatically labeled data can be extracted from the unlabeled data.

For text classification tasks, the evaluation model $Eval$ leverages a variety of features. The confidence score of the main model $Main$s prediction, the n-gram language model related ranking, the input sentence probability scores evaluated by the language model from $Main$s predicted class and the optional noisy user feedback signal about $Main$s output are the features used by $Eval$. The language model related ranking and input sentence probability scores are based on the assumption that sentences belonging to the same class are more similar than sentences belonging to different classes. Thus, we train a language model SLM_i for each $CLASS_i$ using sentences belonging to that class. If a sentence belongs to $CLASS_i$, its sentence probability that is evaluated with SLM_i should be higher than the sentence probability evaluated with $CLASS_j$, where $i \neq j$. We use the trigram statistical language model[3] to train SLM_i for each $CLASS_i$.

Details on how to train the auxiliary evaluation model $Eval$ are described in Algorithm 2. The input to the algorithm is the $Main$ model, the labeled data L and the optional set of user feedback data F. For each example in L, based on l_is labeled class, we create one positive training sample with the correct class and one negative training sample with a wrong class that is chosen randomly. The features can be generated using the aforementioned multiple signal sources, such as Ms prediction confidence scores, the SLM ranking scores and the optional user feedback.

[3]We use KenLM library Heafield (2011) to build and query SLMs.

Algorithm 2 Train Auxiliary Evaluation Model

Given:
$Main$ model
Labeled data $L : \{l_i\}$
Optional feedback data $F : \{f_i\}$
for l_j **in** L **do**
  $CLASS_i$ = class label of l_j
  Append l_j to $SLMCorpus_i$
end for
for i **in** $AllClasses$ **do**
  SLM_i = trainSLM($SLMCorpus_i$)
end for
$Corpus_{xgb}$ = emptyList()
for l_j **in** L **do**
  $CLASS_i$ = class label of l_j
  $XgbSample_+$ = getXgbFeatures(l_j,f_j, M, $SLMs$, $CLASS_i$)
  $XgbSample_-$ = getXgbFeatures(l_j,f_j, M, $SLMs$, $CLASS_k$ where $k \neq i$)
  $Corpus_{xgb}$.append($+$, $XgbSample_+$)
  $Corpus_{xgb}$.append($-$, $XgbSample_-$)
end for
$Eval$ = trainXgbModel($Corpus_{xgb}$)

3 Theoretical Analysis

The EM-like semi-supervised learning approach with an auxiliary evaluation component is designed to tackle large scale ML problems. In section 5, we will demonstrate that our framework has a superior and consistently better performance in various real-world machine learning tasks based on the empirical results. In this section, we will first analyze and highlight some mathematical aspects of this dual-player, semi-supervised learning approach, and illustrate its deep connection to the Expectation-Maximization algorithm.

Suppose we are given an initial set of N manually labeled text $\mathcal{S}^{(0)}$, and our main task is to classify unseen text to a label. As described before, we use a shallow neural network similar to Joulin et al. (2016) to build the $Main$ model. For this

purpose, according to Joulin et al. (2016), we want to minimize the negative log-likelihood

$$-\frac{1}{N}\sum_{1}^{N} y_n log\left(f\left(BAx_n\right)\right) \tag{1}$$

where x_n is the normalized bag of features of the n^{th} text, y_n is the category labels, and A and B are the weight matrices. As part of the auxiliary evaluation component $Eval$, we established a machine learning system with richer context compared to $Main$. The task of $Eval$ is to estimate the probability that the given input text belongs to the category predicted by $Main$. This probability is defined as p_{text_i,c_j}.

$$p_{text_i,c_j} = P(C_j|text_i) \tag{2}$$

Notice that the entire purpose of the evaluation system is to select newly labeled data to enrich the training set of the main machine learning system. Thus, the $Eval$ model estimates the confidence score of this prediction for each sample. The whole learning process of $Main \rightarrow Eval$ iterates as described in the previous sections. The dual system runs iteratively. We stress that it has a close connection to the popular Expectation-Maximization algorithm Dempster et al. (1977) via the following result.

Theory 1. *Given a two-player machine learning system comprised of Main and Eval, the Main model converges to the local minima of the negative log-likelihood with the controlled false positive rate given enough capacity.*

Proof. Given a set of training data $\mathcal{S}^{(0)} = (\mathbf{x}_i, y_i)$, $i = 1, \cdots, N$, which are the observed features and labels, let us denote a hidden variable $z_i \in \{0, 1\}$ that is a variable indicating the quality of the observation. z_i takes a value of 1 if the label for the corresponding instance is correct or relevant, and 0 otherwise. Without the loss of generality, suppose that the $Main$ model is trained to maximize the log-likelihood function:

$$\ell\left(\Theta|\mathbf{X}, \mathbf{Y}\right) \tag{3}$$

Using equation (1), equation (3) can be rewritten as

$$\begin{aligned}
\ell\left(\Theta|\mathbf{X}, \mathbf{Y}\right) &= \log p\left(\mathbf{X}, \mathbf{Y}|\Theta\right) \\
&= \log \sum_{z} p\left(\mathbf{x}, y, z|\Theta\right) \\
&= \log \sum_{z} p(z) \frac{p\left(\mathbf{x}, y, z|\Theta\right)}{p(z)} \\
&\geq \sum_{z} p(z) \log \frac{p\left(\mathbf{x}, y, z|\Theta\right)}{p(z)} \\
&= \mathbf{E}_{p(z)} \log p\left(p(\mathbf{x}, y, z|\Theta\right) \\
&+ Entropy[p(z)] \\
&= L\left(p, \Theta; \mathbf{X}, \mathbf{Y}\right)
\end{aligned} \tag{4}$$

where the inequality is obtained by Jensen's inequality. The equality holds if and only if

$$p(z) = p(z|\mathbf{X}, y, \Theta)$$

The term $\mathbf{E}_{p(z)} \log p\left(p(\mathbf{x}, y, z|\Theta\right)$ is the expected complete log-likelihood (or, **Q-function**). The two machine learning systems then iterate through the following two steps. From a set of noisy data, $Eval$ performs similarly in the **E**-step of the EM algorithm. For n^{th}step, $(n = 1, 2, \cdots)$:

$$\begin{aligned}
p(z)^{(n)} &= \arg\max_{p} L\left(p, \Theta^{(n-1)}; \mathbf{X}, \mathbf{Y}\right) \\
&= p(z|\mathbf{X}, y, \Theta^{(n-1)})
\end{aligned} \tag{5}$$

Notice that the conditional distribution of the hidden variable z is not necessarily fully predictable by the machine learning model even if the observed data and the models parameters are given. The evaluation system mainly provides a confidence score of the correctness or confidence of the prediction, which is defined by equation (5). By properly controlling the false positive rate, we select only those new training examples with a good estimate of $p(z)^{(n)}$ by the $Eval$ model. This results in a set of filtered samples $\mathcal{S}^{(n)}$ to be added to our $Main$ system for the next iteration. The main system then performs the maximization step role in the EM algorithm framework, which is the M-step that follows:

$$\begin{aligned}
\Theta^{(n+1)} &= \arg\max_{\Theta} L\left(p^{(n)}, \Theta; \mathbf{X}, \mathbf{Y}\right) \\
&= \arg\max_{\Theta} \mathbf{E}_{p(z)^{(n)}} \log p\left(\mathbf{x}, y, z|\Theta\right)
\end{aligned} \tag{6}$$

over $\mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \cdots \cup \mathcal{S}^{(n)}$ which is readily solvable from the main machine learning system. Notice that in the **M**-step, it is not necessary to find the optimal values over the whole parameter space. Using the monotonic convergence

property of the generalized EM algorithm, given enough capacity, the $Main$ system would eventually converge to its local optimum after enough iterations. □

In the e-commerce scenario, we have more informative features in the offline $Eval$ system, and thus the evaluation system can have a very high accuracy. According to the proof, the main machine learning system eventually reaches a stable state.

4 Related Work

Various semi-supervised learning approaches have been proposed to leverage unsupervised data to improve the performance of machine learning systems Triguero et al. (2015).

Active learning Cohn et al. (1996); Nigam et al. (1998); Beygelzimer et al. (2009), which is a special kind of semi-supervised learning, provides ways to actively select the most informative data samples from a vast amount of unlabeled data. The selected samples are then labeled by humans. In this way, the total amount of data needed for manual labeling is reduced to save resources. How to handle the problem of label quality is one of the active areas of active learning research. Zhang and Chaudhuri (2015) studied the problem of active learning where labels were obtained from strong and weak labelers. In addition to the standard active learning setting, they consider the problem where they have extra weaker labelers that may provide incorrect labels. Yan et al. (2016) studies the adaptive active learning problem where the labeler can return incorrect labels and also abstain from labeling.

Although active learning can significantly reduce the amount of manual labeling, it still requires extra human labeling, which is costly and time consuming. Compared with active learning, our approach does not require any additional manual labeling effort.

The self-labeled technique is another type of semi-supervised approach to boost the models performance by iteratively labeling parts of the unlabeled data. This approach aims to obtain an enlarged labeled set, which is based on its most confident predictions, to classify unlabeled data. Zhu and Goldberg (2009) divides the self-labeled methods into self-training and co-training.

In the self-training process Triguero et al. (2014); Yarowsky (1995), a model is trained with an initially small number of human labeled examples that aim to predict unlabeled data. Then, it is retrained with its most confident predictions, thus enlarging its labeled training set. The process iterates in the same manner.

In the co-training process Blum and Mitchell (1998); Chen et al. (2011), two learning models are trained separately to provide distinct views of the data set by using different feature sets of the data. These two models are initially trained with a small amount of human labeled data, and then the most confident predictions of one model on the unlabeled data are used to construct the training data for the other model. This process is repeated iteratively. Similar to our proposed approach, the self-labeled method uses the EM-based iterative process to boost the models accuracy and also does not need any further manual labeling efforts.

The major difference between the self-labeled approach and our approach is as follows. In the self-labeled method, with either self-training or co-training, all the models are main task machine learning models. In our proposed approach, there exists only one main-task model and another auxiliary evaluation model that runs offline. Using an offline auxiliary evaluation model has the benefit of utilizing offline information that is not available at prediction time. Thus, the auxiliary evaluation model has a better estimation capability than the main model regarding whether Mains output is correct or not.

The Generative Adversarial Network Goodfellow et al. (2014) is another semi-supervised approach that tries to generate unlimited synthetic fake samples that can mimic real data. The GAN also builds two models, namely, the generative model and the discriminative model, and puts them against each other. The generative model takes random inputs and tries to generate output data that looks similar to real data. The discriminative model takes input data from both the generative model and real data and tries to correctly distinguish between them. The GAN has been successfully applied to image and audio areas where the synthetic data is real-valued. It's quite challenging for the GAN to generate sequence of discrete tokens in the NLP domain. Yu et al. (2017) has proposed the SeqGAN method to address this challenge by directly performing gradient policy update with reinforcement learning. Kusner and Hernández-Lobato (2016) pro-

posed an alternative method to address this challenge using the Gumbel-softmax distribution.

One of the differences between our approach and GAN is that our approach relies on real unlabeled data while the GAN generates plausible data with random inputs. Another major difference is that the evaluation component in our approach tries to evaluate whether the main model results are correct or not. Meanwhile, in the GAN approach, the adversarial component learns to tell whether the current data sample is real or fake.

5 Experiments

To illustrate the effectiveness of the proposed semi-supervised learning method, we evaluate it with different text classification tasks. We compare the new method with a few other benchmark semi-supervised approaches using the public Yahoo! Answers topic classification dataset Zhang et al. (2015); Joulin et al. (2016) and our e-commerce product categorization dataset.

5.1 Yahoo! Answers Dataset Experiments

The Yahoo! Answers topic classification dataset contains 10 classes. Each class contains 140K training samples and 6K testing samples. In this dataset, the total number of training instances is 1.4M and total number of test instances is 60K Zhang et al. (2015). We shuffle and split the original 1.4 M labeled training data into two sets. The first set contains 100K instances with labels and is used as the initial labeled dataset L. The second set contains 1.3 M instances and the labels are deleted to form the unlabeled dataset U. The 60K test samples are untouched as the blind test set T.

5.1.1 Results

Using the initial 100K labeled dataset L and 1.3 M unlabeled dataset U, we compare our approach to the three benchmark approaches: supervised learning, co-training and active-learning. In all experiments, once the labeled training corpus for the main model is derived, we use the shallow neural network classifier described in Joulin et al. (2016) to train $Main$.

1. **1.4M Supervised Learning:** Use the entire Yahoo! dataset and build a model similar to Joulin et al. (2016). The accuracy is reported as 72.3%. This is the theoretical upper bound for a semi-supervised learning training. Any proposed method with less labeled data tries approach this accuracy.

2. **100K Supervised Learning:** Use L dataset and build a model similar to Joulin et al. (2016). The accuracy for this model is 65.9%. This is the lower bound result. Any proposed method to leverage unlabeled data should outperform this number as much as possible.

3. **Co-Training:** Use L to build two initial models and use U data in a co-training setup to enhance the initial models. The system converged to an accuracy of 69.03% after 40 iterations.

4. **Self-Training:** Use L to train an initial model and use this initial model to predict the labels of U. In the next step, mix L and U with its predictions to train a new model. Keep iterating the predicting labels for U, mixing corpus and training the classifier until the system converges. In our experiments, the self-labeled baseline converged after 30 iterations to an accuracy of 67.5%.

5. **Active Learning:** Train the initial classifier L, use it to evaluate and select the most informative samples from U and reveal their ground truth labels. Then, update the classifier with the mixed corpus of L and reveal the label samples. As more samples get selected, the performance of the active-learning algorithm improves. Fig 1 shows the improvements in the accuracy with the increasing amount of manual labeling data.

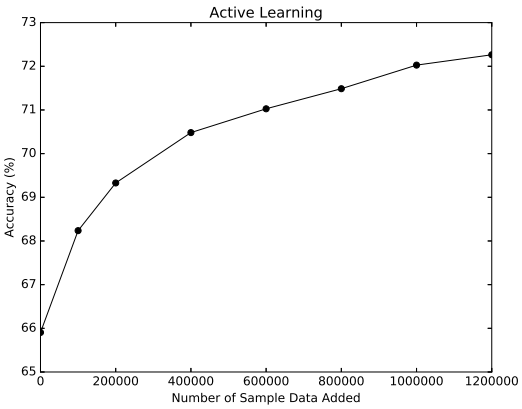

Figure 1: active learning accuracy w.r.t the number of samples selected for labeling

6. **EMAEC without Enriched Data:** Use L and U and apply Algorithm 1 and Algorithm 2 to iteratively train $Main$ and $Eval$. After approximately 20 iterations, $Main$ can achieve an accuracy of 70.42%. The $Eval$ model yields 92.8% precision with 83.5% recall. At the convergence stage, the system generated labels for 1.12M instances in U with a false positive rate of $<8\%$. This approach automatically labels the majority ($>86\%$) of the U dataset with high-quality (error rate $<8\%$).

7. **EMAEC with Enriched Data:** The Yahoo! dataset does not contain any additional user session feedback data. To simulate the scenario where user-provided feedback data is unreliable to produce 100% correct automatic labels, we assume that user feedback data can be simulated by randomly introducing noises to the original ground truth labels in the dataset. Thus, we first reveal all the ground truth labels in U, and then randomly select $x\%$ of U. Next, we randomly flip their correct labels into wrong labels and then mix them with the remaining instances in U. We call the mixed and blurred dataset as B which is the U dataset with noisy labels. We use the B dataset to simulate user noisy feedback signals. In this experiment, we use L, B, Algorithm 1 and Algorithm 2 to iteratively train $Main$ and $Eval$. As expected, the higher that the level of blurring is, the worse that the system performs. The theoretical upper bound for this experiment would be a classifier trained with 100K + $(1 - x\%)$ * 1.3M ground truth labeled data. Figure 2 demonstrates the varying system performance varying with different noise level $x\%$. We can see that user feedback loop data can further improve the system's performance even if we introduce 50% noise to B.

5.1.2 Discussion

The experimental results on the Yahoo! Answers topic classification dataset are summarized in Table 1. The results demonstrate that by starting with only 100K labeled data and 1.3 M unlabeled data, and by using an auxiliary evaluation component, the systems accuracy can be increased from 65.9% to 70.4%. Active learning can reach the same performance only after adding another 400K manually labeled data.

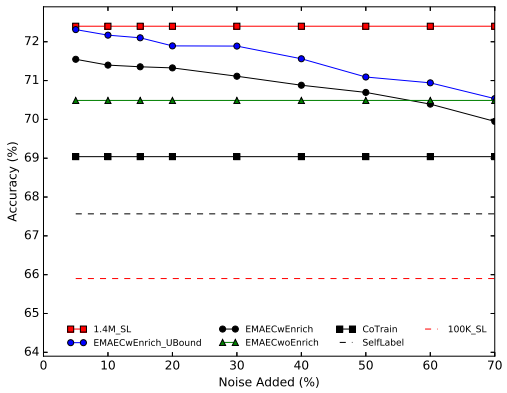

Figure 2: EMAEC with enriched data at different noise levels and the comparison with other baselines.

Approach	Accuracy [%]	Error Reduction [%]
100K Supervised Learning	65.90	0.000
Self-Training	67.57	4.888
Active Learning (extra 100K labeled)	68.23	6.85
Co-Training	69.04	9.208
Active Learning (extra 400K labeled)	70.48	13.436
EMAEC w/o Enriched Data	70.49	13.456
EMAEC w/ Enriched Data (20% noise)	71.33	15.913
Supervised Learning with 20% noise	57.8	-12.29
1.4M Supervised Learning	72.40	19.062

Table 1: Test accuracy and error reduction rate [%] on the Yahoo! Answers dataset. The method proposed in this study is printed in bold.

Moreover, the proposed approach can automatically generate high-quality labels for over 86% of the unlabeled data with an error rate less than 8%.

The results also show that by adding simulated user feedback loop signals into the evaluation component, the final system accuracy can be further improved. Even with 50% label noise, the system achieves 71.33% accuracy. The active learning system can achieve the same accuracy only after adding extra 800K manually labeled data. It's also worth noting that with the same noisy blurred label dataset, the supervised learning approach has much worse performance. Its classifier accuracy significantly drops to 57.8% with 100K golden label and 1.3M blurred labels at 20% noise level.

5.2 E-commerce Product Categorization Dataset Experiments

The proposed method is derived to tackle large scale text classification problems that occur in the e-Commerce industry, where the challenge is that

we *significantly* lacked high-quality labeled data for these problems. For example, the e-commerce product categorization dataset contains product titles and 600 different categories for the product titles. This dataset contains four different parts: the product category description data for 600 categories, a 6K observation manually labeled initial training dataset L, a 28K observation manually labeled blind test set T, and a 3.5 million observation unlabeled dataset U that included rich user feedback session data F. The main task here is to predict the product category as soon as the online user enters the product title. For example, the user might enter a product title, such as green coach bag to describe a product. The system should classify this input title into the most relevant product category, such as "women's purse & bag". The 3.5 million unlabeled user behavior dataset contains a seller chosen category and a category suggested by a machine learning model. We consider these data to be unlabeled since the seller chosen category has a greater than 30% error rate according to our evaluations. The reason for this high error rate is due to the fact that the users are not familiar with the category tree or they just intentionally select the wrong category to increase the chance of selling their product. Note that for the main-task system that runs online, only the product title information is available to main-task model.

5.2.1 Results

With the initial 6K labeled dataset L, the 3.5M unlabeled dataset U and the 3.5M feedback session dataset F, we compare our proposed EMAEC approach with the auxiliary evaluation component to a weak supervised learning baseline and co-training baseline as described below. Similar to the previous set of experiments, once the labeled training corpus for the main model is derived, we use the shallow neural network classifier described in Joulin et al. (2016) to train $Main$.

1. **Supervised Learning with Small Labeled Data:** We train a supervised baseline model with the 6K labeled dataset L. This is the weak baseline model. Any proposed method that leverages unlabeled data U and session data F should outperform this number.

2. **Supervised Learning with Noisy Data:** We treat the 3.5M seller-chosen category as the correct labeled data from F, and then mix it with the 6K labeled dataset L to train a supervised model. Total error reduction rate by adding seller-chosen labels is 12.3%

3. **Co-Training:** Use L to build two initial models and use U data in a co-training setup to enhance the initial models. Different feature sets from F are used to train two different models. After approximately 30 iterations, the system will converge to best performance. Total error reduction rate for co-training is 15.2%.

4. **EMAEC with Enriched Data:** Build the initial classifier $Main$ using L and F. Apply Algorithm 1 and Algorithm 2 to iteratively train the $Main$ and $Eval$ models. After approximately 20 iterations, the main task classifier $Main$ converges to its best performance. Total error reduction for our approach is 19.23%.

Model	Error Reduction Rate [%]
Supervised Learning with 6K Label Data	0.00
Supervised Learning with Noisy Data	12.30
Co-Training	15.20
EMAEC	19.23

Table 2: EMAEC gain in error reduction rate [%] compared to the co-training baseline on the e-commerce dataset

5.2.2 Discussions

The experiment results on the e-commerce product categorization dataset are summarized in table 2. The results demonstrate that our proposed approach with the auxiliary evaluation component outperforms the co-training approach substantially. The classification error rate is reduced by 5%. This improvement is well aligned with the results on the public Yahoo! Answers dataset.

6 Conclusions

In this paper, we proposed a semi-supervised learning approach to tackle the challenge of lacking high-quality labeled data. The experimental results in text classification tasks with both open source Yahoo! Answer data and our e-commerce data show the effectiveness of the proposed approach. This general dual player machine learning framework can also be applied to other machine learning tasks, such as search ranking, speech recognition, machine translation, etc.

The proposed method comes with advantages and disadvantages over existing semi-supervised learning approaches. The advantages have been

demonstrated in text classification tasks in that it can automatically extract fairly high-quality predicted labeled data from massive unlabeled data. Thus, it can further improve prediction accuracy by adding those automatically enriched labeled data into the original training corpus.

On the other side, a potential drawback could be that its effectiveness may be limited by the prediction performance of the auxiliary evaluation model. If the auxiliary evaluation model is not able to generate many labeled samples with low false positive rate, the automatically enriched labeled data might not be well distributed to reflect the real problems underlying data distribution. To overcome this, we must rely on vast amounts of real-world unlabeled data.

At last, we know that the GAN based approach Goodfellow et al. (2014); Yu et al. (2017); and Kusner and Hernández-Lobato (2016) can automatically generate infinite amounts of fake data. Combining the advantages of the GAN framework and the proposed approach is a very interesting research direction for us in the future.

References

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56. ACM.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *NIPS*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, pages 2672–2680.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv:1607.01759*.

Matt J Kusner and José Miguel Hernández-Lobato. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom Mitchell, et al. 1998. Learning to classify text from labeled and unlabeled documents. *AAAI/IAAI*, 792.

Isaac Triguero, Salvador Garca, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*.

Isaac Triguero, José A Sáez, Julián Luengo, Salvador García, and Francisco Herrera. 2014. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big Data*, 3(1):1–40.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2016. Active learning from imperfect labelers. In *Advances in Neural Information Processing Systems*, pages 2128–2136.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

Chicheng Zhang and Kamalika Chaudhuri. 2015. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pages 703–711.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NIPS*.

Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.